Problem solving and practice: C++ Assignment:2

Student ID: 18013189

Department: Computer Engineering(컴퓨터공학과)

Name: Cha yun beom(차윤범)

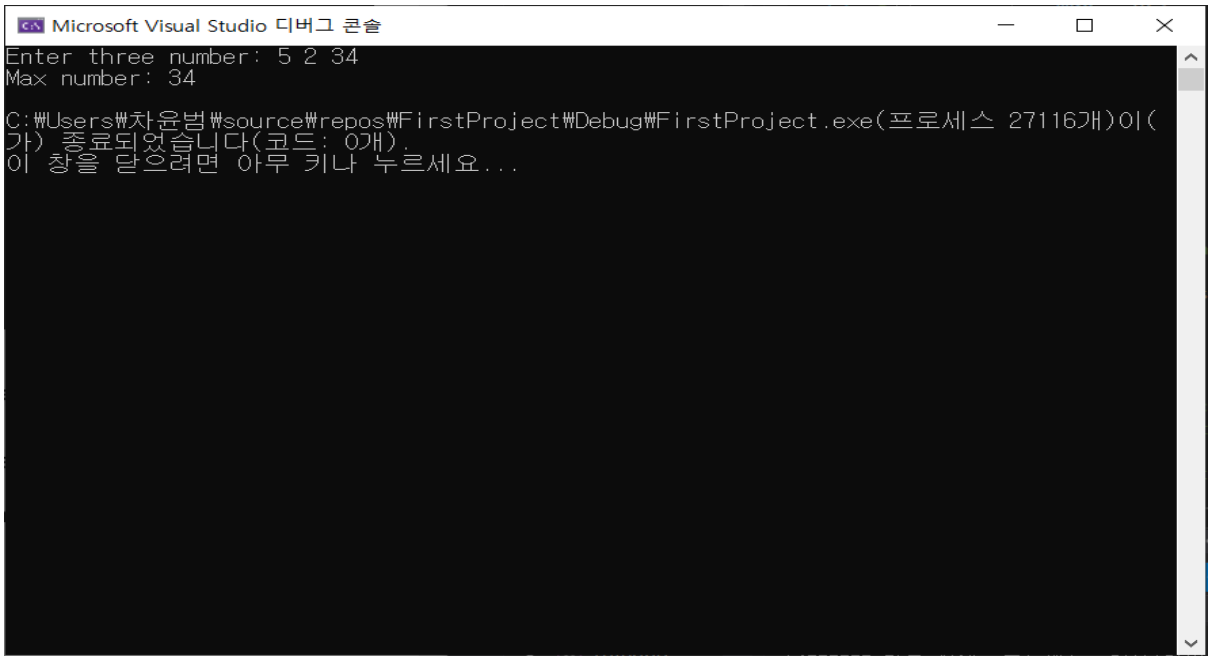
Problem1

```
#pragma warning(disable:4996)
#include<iostream>
#include<string>
#include<cstring>
#include<cstdlib>
#include<ctime>

using namespace std;

class largest { //class definition
    int n1, n2, n3;
public:
      //method declare
    void input();
    int max();
    void display();

};
//inline function definition
inline void largest::input() {
    cout << "Enter three number: ";
    cin >> n1 >> n2 >> n3;
}
inline int largest::max() {
    int max = n1;
    //max
    if (max < n2){
        if (n2 > n3){
            max = n2;
        }
        else{
            max = n3;
```

```
        }
    }
    else{
        if (max < n3){
            max = n3;
        }
    }
    return max; //return value
}
inline void largest::display() {
    cout << "Max number: " << max() << endl; //output message
}

int main() {
    largest big; //object as big
      //function call
    big.input(); //input value
    big.max(); //max
    big.display(); //output value
    return 0;
}
```

The method was declared by defining the target class, and inline functions were defined. Input() is designed to receive three numbers, and max is designed to compare the three numbers and return the largest number.
The main function declares the target class object to invoke the function.

Problem1 output screen



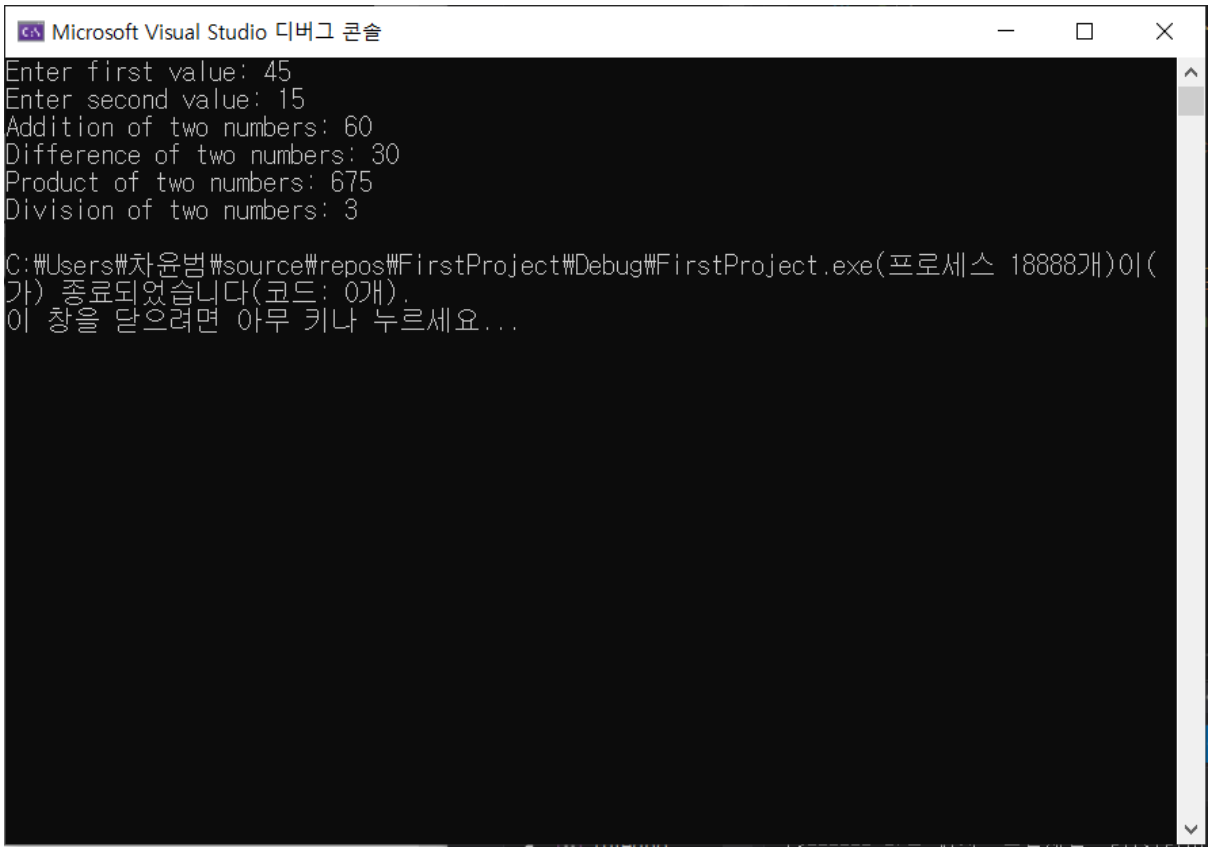
Problem2

```
#pragma warning(disable:4996)
#include<iostream>
#include<string>
#include<cstring>
#include<cstdlib>
#include<ctime>

using namespace std;

class calc { //class calc definition
    int n1, n2;
public:
    //method declare
    void get();
    void sum();
    void difference();
    void product();
    void division();
};

//inline function definition
inline void calc::get() {
    cout << "Enter first value: ";
    cin >> n1;
    cout << "Enter second value: ";
    cin >> n2;
}
inline void calc::sum() {
    cout << "Addition of two numbers: " << n1 + n2 << endl; //Add op
eration
}
inline void calc::difference() {
    cout << "Difference of two numbers: " << n1 - n2 << endl;
    //Sub operation
}
inline void calc::product() {
    cout << "Product of two numbers: " << n1 * n2 << endl;
    //Mutiply operation
}
```

```
inline void calc::division() {
    cout << "Division of two numbers: " << n1 / n2 << endl;
    //Division operation
}
int main() {
    calc obj1; //object as obj1
    //function call
    obj1.get();
    obj1.sum();
    obj1.difference();
    obj1.product();
    obj1.division();

    return 0;
}
```

I defined a calc class and have declared a method.
The get() function was declared to receive two numbers, and the sum(), difference(), product(), and division() functions were designed to output the two numbers received by +,-,*,/ calculating.
The main is designed to declare objects in the calc class to call and output each function.

Problem 2 output screen



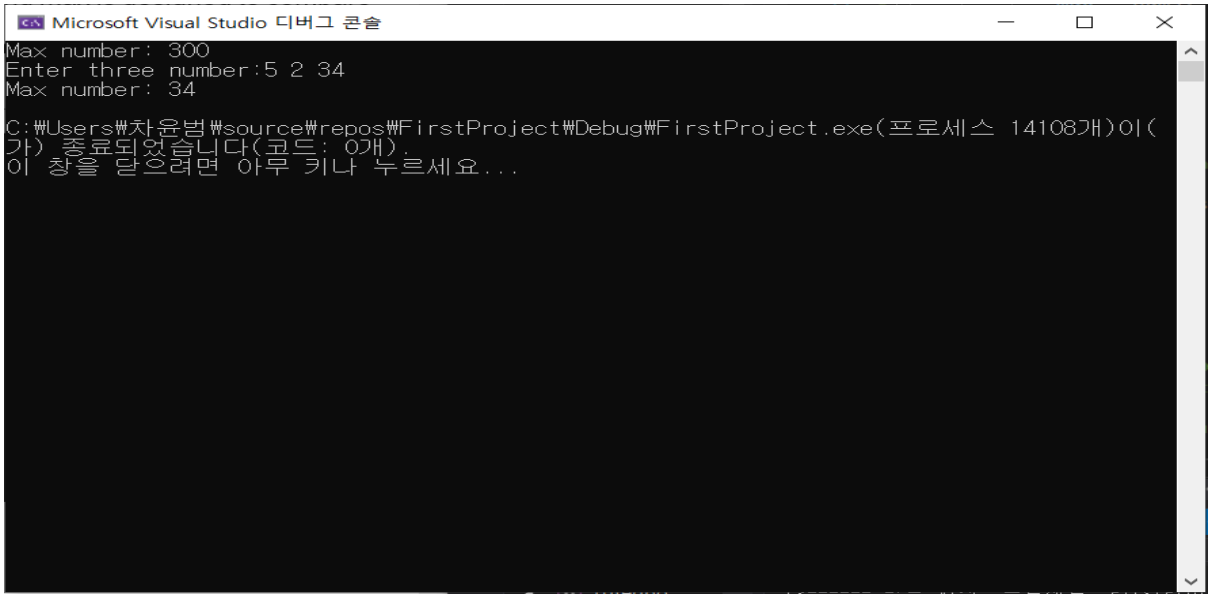
Problem3

```
#pragma warning(disable:4996)
#include<iostream>
#include<string>
#include<cstring>
#include<cstdlib>
#include<ctime>

using namespace std;

class largest {
    int num1, num2, num3;
public:
    largest(int x, int y, int z) {
        num1 = x;
        num2 = y;
        num3 = z;
    }

    int max();
    void display();
    void input();
};
inline void largest::input() { //input value
    cout << "Enter three number:";
    cin >> num1 >> num2 >> num3;
}
inline int largest::max() {
    //max value
    int max = num1;
    if (max < num2) {
        if (num2 > num3) {
            max = num2;
        }
        else {
            max = num3;
        }
    }
    else {
        if (max < num3) {
```

```
            max = num3;
        }
    }
    return max; //return value
}
inline void largest::display() {
    int large = max();
    cout << "Max number: " << large << endl; //output message
}

int main() {
    largest big(10, 30, 300); //parametrized contructor 3 arguments
    //function call
    big.display(); //(10,30, 300) argument output
    big.input(); //input value
    big.max(); // max
    big.display(); // Output the value inputed
    return 0;
}
```

The method was declared by defining the target class. Designed by declaring a constructor that receives three arguments within the class.
The input() function is designed to receive three numbers, and max() compares the three numbers received and returns the largest number. display() causes the returned large number to be output.
The main declared object of the target class with arguments 10, 30, 300.
The function is called to output the largest value for 10, 30, 300, and the largest number by receiving the three numbers below.

Problem 3 output screen



Problem4

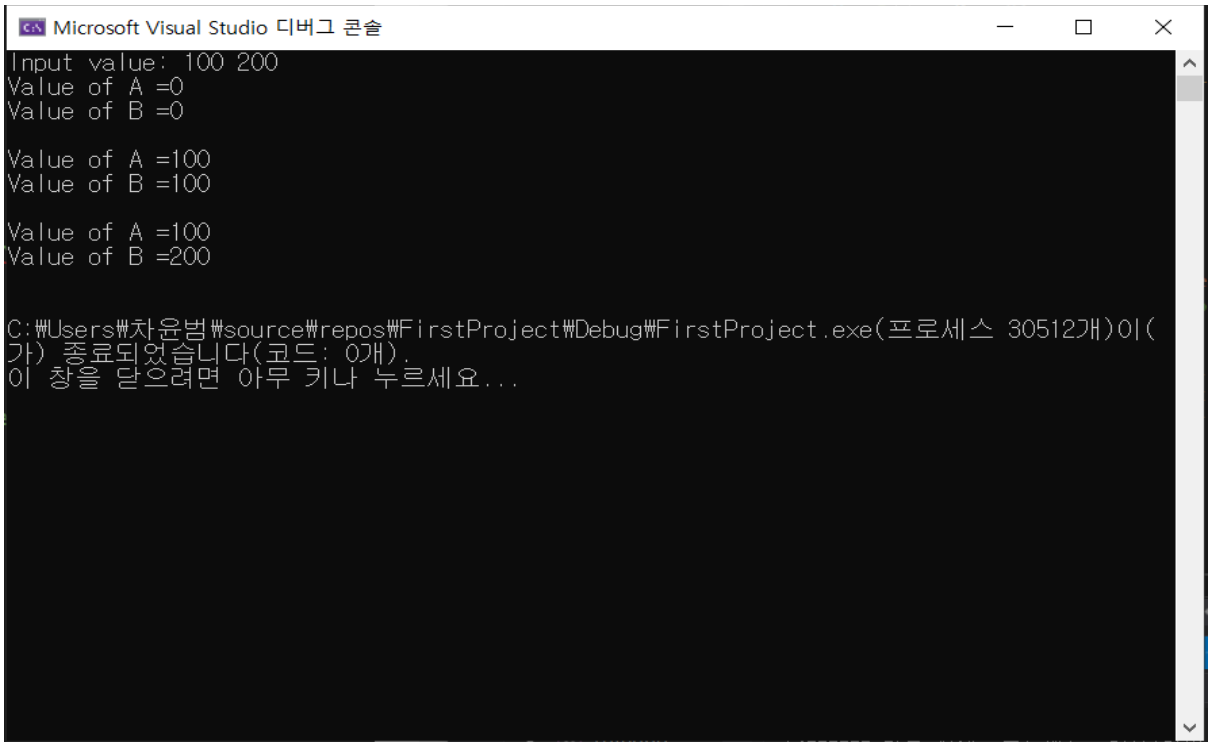
```
#pragma warning(disable:4996)
#include<iostream>
#include<string>
#include<cstring>
#include<cstdlib>
#include<ctime>

using namespace std;

class Arguments_Print { //class Arguments_Print definition
    int a, b;
public:
    //default constructor
    Arguments_Print();
    //parametrized constructor single argument
    Arguments_Print(int num);
    Arguments_Print(int x, int y);
    void display();
};
//default constructor
Arguments_Print::Arguments_Print() {
    a = 0;
    b = 0;
}
//parametrized constructor single argument
Arguments_Print::Arguments_Print(int num) {
    a = b = num;
}
//parametrized constructor single argument
Arguments_Print::Arguments_Print(int x, int y) {
    a = x;
    b = y;
}
void Arguments_Print::display() {
    cout << "Value of A = " << a << endl;
    cout << "Value of B = " << b << endl;
    cout << endl;
}
int main() {
```

```
    int a, b;
    cout << "Input value: ";
    cin >> a >> b; //input value

    Arguments_Print ap; //default argument
    Arguments_Print ap1(a); //1 argument
    Arguments_Print ap2(a, b); //2 argument
    //output message
    ap.display();
    ap1.display();
    ap2.display();

    return 0;
}
```

I defined the Argument_Print class, declared a constructor with two or one default and a display() method.
The default constructor has initialized to zero.
The constructor with 1 argument has the same value as a = b = num.
The constructors with two instruments put values a = x and b = y.

The main function is designed to output two values and an object from Argument_Print by declaring each class constructor.

Problem 4 output screen



Problem5

Main.cpp

```
//That stores the application, i.e. the main() method for the applic
ation.
#pragma warning(disable:4996)
#include<iostream>
#include<string>
#include<cstring>
#include<cstdlib>
#include<ctime>
#include "Class.h" //Declare header files

using namespace std;

int main() {
    largest big; //object in class.h as big
    //function call
    big.input();
    big.max();
    big.display();
    return 0;
}
```

Class.cpp

```
//That stores the method definitions for that class.
#pragma warning(disable:4996)
#include<iostream>
#include<string>
#include<cstring>
#include<cstdlib>
#include<ctime>
#include "Class.h" //Declare header files

using namespace std;

void largest::input() {
    cout << "Enter three number: ";
    cin >> n1 >> n2 >> n3; //input value
```

```
}
int largest::max() {
    //max
    int max = n1;
    if (max < n2) {
        if (n2 > n3) {
            max = n2;
        }
        else {
            max = n3;
        }
    }
    else {
        if (max < n3) {
            max = n3;
        }
    }
    return max;
}
void largest::display() {
    cout << "Max number: " << max() << endl; //output message
}
```

Class.h

```
//That stores the class declaration and definition.
#pragma once
#ifndef CLASS_H //Conditional processing
#define CLASS_H
class largest {
    int n1, n2, n3;
public:
    //method declare
    void input();
    int max();
    void display();
};
#endif
```

A class file was created and declared to main.cpp as #include "Class.h" to declare object big of the largest class. The function is called below.
Class.cpp similarly defined an inline function of the largest class by declaring #include "Class.h". (Inline function is the same as Problem 1)
Class.h defined the target class to declare each method and conditioned it with #ifndef and #endif.

Problem 5 output screen